Linear Search

If we want to determine if an item is in a list we must do a search. The first searching technique we will look at the is the Linear Searching technique.

The Linear Searching technique is good to use when the data in the list is in no particular order. To do a linear search, start at one end and look at consecutive locations until you find the "key" (the key is the time you are looking for) or you searched the entire list. For example:

```
struct DataType
{
    int info;
    int key;
};

typedef apvector<DataType> List;

int LinearSearch(const List & x, int key)
// pre: x is a list containing unordered data
// post: return the index of where key is found
//       return -1 if key is not found
{
    for (int i = 0; i < x.length(); i++)
        if (x[i] == key)
            return i;
    return -1;
}
```

The least number of searches this algorithm would take to find a key is 1. That is it would find the key in the first location of the list.

The most number of searches this algorithm would take to find the key is *n*. The variable *n* is used to indicate the number of elements in the list, in this case it's value is *x.length()*.

The average number of searches this algorithm would take to find the key is $\frac{n}{2}$.